

BEYOND THE BACKLOG

The Backlog Was Always a Rationing Device — and Autonomous Agents Just Removed the Scarcity

When execution becomes free, the queue stops being the point — and delivery's centre of gravity shifts upstream

Giovanni Leonardi

June 2024



The Backlog Was Always a Rationing Device — and Autonomous Agents Just Removed the Scarcity

Transformation · No. 383

Originally written June 2024 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

"You can automate the queue; you cannot automate the responsibility for what is in it."

1. When the Queue Stops Being the Point

The demonstration always lands the same way. A delivery team gathers around a screen while an agent works through the backlog — not a contrived example, but their own sprint, the one they had committed to on the Monday. By mid-afternoon it has opened, edited and proposed finished changes for the better part of two weeks' planned work. There is a round of applause, and then something quieter: a room full of experienced people looking at one another with the particular unease of professionals who have just watched a familiar instrument turn strange in their hands.

The instinct in that moment is to reach for the productivity story. If an agent can clear a sprint in an afternoon, we will simply ship more, and ship it faster. That story is not so much wrong as it is answering the wrong question. What has actually happened in that room is not that the work got faster. It is that the backlog — the ordered list of things-to-be-done that has sat at the centre of how we manage delivery for the better part of twenty years — has quietly lost the job it was really doing.

This is a piece about what that job was, why autonomous execution dissolves it, and why the comfortable reading — the agents clear the queue, so we go faster — is the most expensive mistake on the table right now.

2. The backlog was a rationing device

We tell ourselves the backlog is a list of work. It is more honest to say it was a rationing device.

For as long as software has been built by teams, the binding constraint has been delivery capacity. There were always more things worth doing than there were hands to do them, and the backlog was the mechanism by which we made that scarcity bearable. It quietly did three jobs at once, and we rarely bothered to separate them:

- It rationed scarce capacity. The ordered list was a queue precisely because the server behind it was slow and expensive. Prioritisation only means something when you cannot do everything.
- It was a negotiation surface. The backlog was where demand — stakeholders, customers, the market — argued with supply, the team's finite throughput. It is where "everything is urgent" met "we can do six things this quarter" and a compromise got struck in public.
- It was a container for intent. A refined item is a small contract: someone has thought about what is wanted, why it is wanted, and how we will know it is done.

The craft we built around the backlog — refinement, estimation, velocity, the sprint commitment — was almost entirely craft for managing the first of those jobs. Story points do not measure value; they measure cost against a scarce resource. Velocity is a throughput number for a constrained system. Burn-down charts assume there is something to burn down against a fixed rate of work. The entire apparatus takes it for granted that the bottleneck is us.

3. What autonomous execution actually removes

Now hold that apparatus up against an agent that can take a well-specified item and execute it at a marginal cost approaching zero.

The thing to notice is narrow but decisive. Autonomous execution does not make the backlog faster. It removes the constraint the backlog existed to manage.

When execution was scarce and expensive, the queue was the whole point — the discipline of deciding what to do first mattered because doing anything at all was costly. When execution becomes cheap and abundant, a queue of executable items is just a list of things that are about to be done anyway. The rationing function — the one all our ceremonies were built to protect — stops binding. And a rationing device applied to something that is no longer scarce does not become more efficient. It becomes theatre.

The backlog is not dying because agents empty it. It is dying because the scarcity it was invented to manage has moved somewhere else.

Where has it moved? To the two jobs we had quietly been getting for free all along: specification and judgement. An agent will faithfully execute a badly chosen item as readily as a well chosen one. It will build the wrong thing beautifully, on time, and to spec. The instant execution stops being the bottleneck, the quality of what you feed the machine — the precision of the intent, the wisdom of the prioritisation — becomes the entire game. The constraint has not disappeared. It has migrated upstream, from can we build it to should we, and do we even know what "it" is.

4. The misreading that looks like success

Here is the trap, and it is worth naming carefully precisely because it wears the costume of success.

Picture a team — composite, but you will recognise it — that adopts agents in earnest. In the old world they closed perhaps eight or ten backlog items in a fortnight. Now an agent returns finished, reviewable work for fifty in the same window. Six weeks in, the burn-down charts are the best anyone has seen; the backlog that used to represent a year of runway is projected to empty by the autumn. Everyone with a dashboard is delighted.

Three months in, someone finally asks a different question: of the two hundred items we have now cleared, how many moved a metric anyone outside this team would recognise? The honest answer is a handful. Not because the work was done badly — it was done immaculately — but because nobody upstream had gone back and re-decided whether those two hundred items were still the right two hundred. They had been written, sized and sequenced for a slow team. The team was no longer slow. And so the organisation had industrialised the delivery of a backlog that was scoped, in every line, for a constraint that no longer existed.

The cost of that mistake appears on no velocity chart. It is visible only in the widening distance between what shipped and what mattered — and by the time that distance is obvious, a quarter has gone.

5. The strongest version of the other side

The serious objection deserves stating at full strength, not as a straw man. It runs like this: we have heard "everything changes" before. Higher-level languages, integrated development environments, continuous integration, the open-source commons — each made execution dramatically cheaper, and yet the backlog survived every one of them. Agents are simply the next entry in a long line of tools that made us faster. The queue endures because the queue is just how humans organise more-work-than-time, and that condition is permanent.

It is a good argument, and it is wrong in one specific way. Every previous gain made a step inside a human-driven flow cheaper. A person still picked up the item, held it in their head, and finished it. That is why the item survived: the backlog item is a human ergonomic. It exists at the size it does — hand-sized, sprint-shaped, estimable — because that is the right size for a person to pick up and complete. An agent has no such ergonomic. It does not need work decomposed into person-sized, two-week-shaped pieces. The moment the consumer of the queue is not a person, the entire grammar of the queue — its granularity, its estimation, its very item-ness — is optimised for a worker who is no longer doing the work.

There is a second objection, subtler: fine, but the agents will simply manage the backlog themselves — groom it, estimate it, sequence it. They can, and they will. But that concedes the point rather than rebutting it. An agent prioritising a backlog is still prioritising against a model of value it did not author. Someone must decide what "valuable" means here, adjudicate the trade-offs that no optimisation function can settle, and answer for the consequences when the call was wrong. You can automate the queue; you cannot automate the responsibility for what is in it. The decision rights are precisely

the part that does not compress.

6. What actually replaces it

The backlog does not vanish into nothing. Its function migrates, and it is worth being concrete about where.

The primary artefact stops being the item and becomes the intent — a clear, verifiable statement of the outcome wanted and the test by which we will know we have it. Specification becomes the craft that estimation used to be. The scarce human act becomes selection and verification: choosing what is worth doing from an effectively unlimited menu, and then checking, hard, that what came back is actually right — because cheap execution also means the cheap and abundant production of output that is plausible and wrong. And the cadence changes. The two-week sprint was a metronome for human throughput; when throughput is no longer the binding constraint, that rhythm is timing a clock that governs nothing. The cadence that matters becomes how often we can re-decide — how quickly intent can be reformed as a fast execution loop teaches us something we did not know when we set out.

	The backlog era	The autonomous era
Binding constraint	Delivery capacity	Specification and judgement
The scarce human act	Estimating and sequencing work	Specifying and selecting work
The core unit	The person-sized item	The verifiable outcome
Cadence set by	Team throughput	Speed of re-deciding
The dominant risk	Not shipping enough	Shipping the wrong thing, efficiently

“The backlog served us honestly for twenty years; its passing is not the loss of a tool but the promotion of the problem.”

The teams that will struggle in the next few years are not the ones that adopt agents slowly. They are the ones that adopt agents quickly and keep their backlogs intact — running a new engine against the old map, industrialising the delivery of work that was scoped, line by line, for a world that has already gone. The question in software was never really how much we can build; that was only ever a constraint, never a purpose. What remains when the constraint lifts — what is worth building, and whether we can recognise it when we see it — is the one thing a full backlog always let us avoid answering directly. It is now the only question left.